The Threat of Adversarial Attacks on Machine Learning in Network Security - A Survey

Olakunle Ibitoye, Rana Abou-Khamis, Ashraf Matrawy and M. Omair Shafiq
School of Information Technology, Carleton University, Ottawa, Canada
Email: {Kunle.Ibitoye, Rana.Aboukhamis, Ashraf.Matrawy, Omair.Shafiq}@carleton.ca

Abstract— **Machine learning models have made many decision support systems to be faster, more accurate and more efficient. However, applications of machine learning in network security face more disproportionate threat of active adversarial attacks compared to other domains. This is because machine learning applications in network security such as malware detection, intrusion detection, and spam filtering are by themselves adversarial in nature. In what could be considered an arms race between attackers and defenders, adversaries constantly probe machine learning systems with inputs which are explicitly designed to bypass the system and induce a wrong prediction. In this survey, we first provide a taxonomy of machine learning techniques, styles, and algorithms. We then introduce a classification of machine learning in network security applications. Next, we examine various adversarial attacks against machine learning in network security and introduce two classification approaches for adversarial attacks in network security. First, we classify adversarial attacks in network security based on a taxonomy of network security applications. Secondly, we categorize adversarial attacks in network security into a problem space vs. feature space dimensional classification model. We then analyze the various defenses against adversarial attacks on machine learning-based network security applications. We conclude by introducing an adversarial risk model and evaluate several existing adversarial attacks against machine learning in network security using the risk model. We also identify where each attack classification resides within the adversarial risk model.**

Keywords: Machine Learning, Adversarial examples, Network security

I. Introduction

There has been an ever-increasing application of machine learning and deep learning techniques in network security. It, however, introduces a new challenge since security and robustness of these models is usually not a huge consideration for machine learning algorithm designers who are more focused on designing effective and efficient models. This creates room for various forms of attack models against machine learning-based network security applications.

Researchers [1][2][3][4] have shown that the presence of adversarial examples can easily fool machine learning systems. Adversarial examples are specially crafted inputs that cause a machine learning model to classify an input wrongly. Machine learning systems typically take in input data in two distinct phases. The training data which is fed into the learning algorithm during the training phase, and the new or test data which is fed into the learned model during the prediction phase. If the attacker can manipulate the input data in either phase, it is possible to induce a wrong prediction from the machine learning model.

In this survey, we provide a brief introduction to machine learning using a three-dimensional classification method. We classify the various machine learning techniques based on the learning tasks, learning styles and learning depth. We further organize the various applications of machine learning in network security based on a taxonomy of security tasks. Next, we classify the various adversarial attacks based on the applications in network security. We identify five main categories of machine learning applications in network security for our classification method. Finally, we classify adversarial attacks against machine learning based on a taxonomy of network security applications.

Our **contribution** is threefold. First, we introduce a new method for classifying adversarial attacks in network security based on a taxonomy of network security applications. We also introduce the concept of problem space and feature space dimensional classification of adversarial attacks in network security.

Secondly, we introduce the concept of adversarial risk in computer and network security. We provide a new model for evaluating the risk of adversarial attacks in network security based on the discriminative or directive autonomy of the machine learning tasks and styles respectively.

Lastly, we evaluate several adversarial attacks against machine learning in network security applications as proposed by various researchers and classify the attacks based on an adversarial threat model taxonomy.

To the best of our knowledge, there is currently no prior work that has reviewed adversarial attacks in network security based on a classification of network security applications. No prior work has also reviewed the concept of problem space vs. feature space dimensional classification of adversarial attacks in network security. Also, this is the first work to propose an adversarial machine learning risk model in the field of network security based on the directive or discriminative autonomy of the machine learning algorithms.

II. Related Work

Adversarial attacks have been widely studied in the field of computer vision [13][14][15] with several attack methods and techniques developed mostly for image recognition tasks. Researchers have discussed the public safety concern

TABLE I
SUMMARY OF RELATED WORK

Reference	Year	Summary
Buczak et al. [5]	2015	Survey focused on complexity and challenges of machine learning based cybersecurity intrusion detection
Gardiner et al. [6]	2016	On the security of machine learning in malware C&C detection: A Survey
Liu et al. [7]	2018	A survey on security threats and defensive techniques of machine learning: A data driven view
Duddu et al. [8]	2018	A survey of adversarial machine learning in cyberwarfare
Akhtar et al. [9]	2018	A survey of adversarial attacks against deep learning in computer vision
Biggio and Roli [10]	2018	Provided an historical timeline of adversarial machine learning over a 10 year period
Zhang et al. [11]	2019	Discussed adversarial attacks as a limitation of deep learning in mobile and wireless networking.
Qui et al. [12]	2019	Generalized survey of adversarial attacks in with brief reference to cloud security, malware detection and intrusion detection.

of adversarial attacks such as in self-driving cars which could be fooled into mis-classifying a stop sign resulting in a potentially fatal outcome [16]. In network security, the consequences of adversarial attacks are equally significant [17] especially in areas such as intrusion detection [18] and malware detection [19] where there have been rapid progress in the adoption of machine learning for such tasks. Even though adversarial machine learning has recently been widely researched in network security, to the best of our knowledge, there is currently no publication that has surveyed the vast number of growing research work on adversarial machine learning in this field. Some existing survey papers we reviewed include Akhtar et al. [9] which reviewed adversarial attacks against deep learning in computer vision. Qui et al [12] provided a generalized survey on adversarial attacks in artificial intelligence, with a brief discussion on cloud security, malware detection and intrusion detection. Liu et al. [7] reviewed security threats and corresponding defensive techniques of machine learning focusing on the threats in the learning algorithms. Duddu el al. in [8] discussed various research work on adversarial machine learning in cyberwarfare, with some mention of adversarial attacks against malware classifiers. Zhang et al. [11] discussed adversarial attacks as a limitation of deep learning in mobile and wireless networking but did not consider deep learning in the context of network security applications. Buczak et al. [5] in their survey on machine learning-based cybersecurity intrusion detection focused on complexity and challenges of machine learning in cybersecurity but did not review adversarial attacks in their study. Biggio and Roli [10] provided an historical timeline of adversarial machine learning in the context of computer vision and cybersecurity but their work did not provide a detailed review in the context of network security. Gardiner et al. [6] in their survey on the security of machine learning in malware detection, focused on reviewing the Call and Control (C & C) detection techniques. They also identified what the weaknesses were and explained the limitations of secure machine learning algorithms in malware detection systems.

None of these previous survey papers shown in Table I has explored the vast amount of research work currently ongoing on the topic of adversarial machine learning in network security in a manner that categorizes them based on security applications, problem or feature space dimensional classification and adversarial risk modelling.

III. Taxonomy and Background

Machine learning enables computers learn to solve specific tasks and make predictions based on past observations [20]. Machine learning algorithms vary significantly, and can be grouped by either task similarity in performing functions, the learning style or the depth of learning. This is illustrated in Figure 1.

A. Machine Learning Styles

We classify machine learning algorithms based on the style in which the model is trained with data. The learning style of the machine learning algorithm has a direct relationship with the directive autonomy of the model discussed in section VI.

1) Supervised Learning: In supervised machine learning, the model learns from a training dataset that consists of a labeled input and desired output pairs. It generates a mapping function that maps between the input (x) and output (y) by analyzing the training dataset to produce a mapping function [21]. Typical applications of supervised learning are for Regression and Classification tasks.

2) Unsupervised Learning: For certain applications where a labelled dataset is not readily available, a different approach to learning is required. Unsupervised learning styles train a model without providing a labeled input or any output variable to be predicted [22]. Unsupervised learning may be used for clustering some input data based on the information and characteristic of the data. Dimensionality Reduction and Association Rule Learning are typical applications of unsupervised learning.

3) Semi-Supervised Learning: In semi-supervised learning, a large amount of unlabeled data with labeled data is used to achieve a better classifier model [23]. Usually, classifiers are trained by using labeled data that consist of input and output pairs and features. Collecting labeled data is often hard, expensive, time-consuming and requires experienced user input [23]. Unlabeled data is easy to collect, but they are limited in terms of usage. Examples of tasks that make use of semi-supervised learning include Regression and Classification.

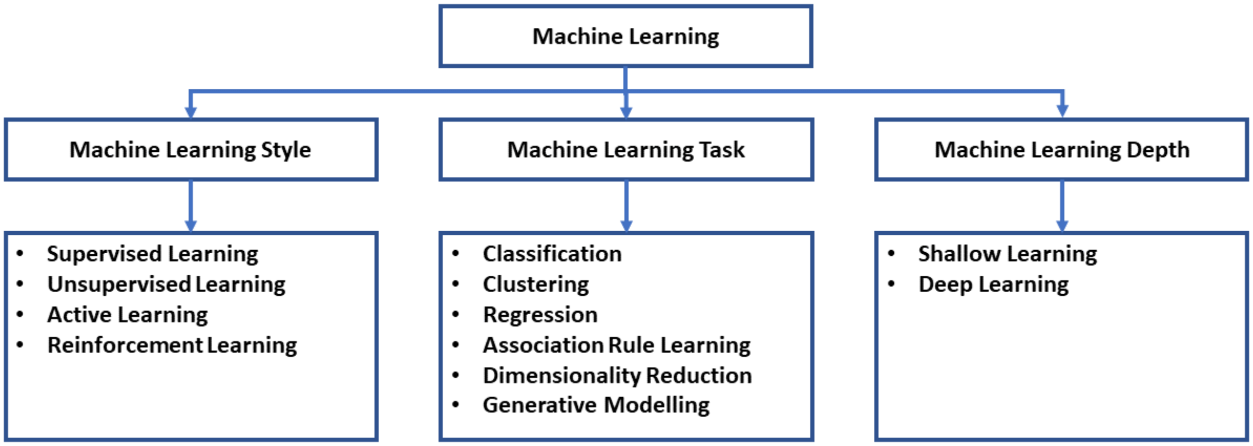

Fig. 1. Three Dimensions of Machine Learning Classification

4) Active Learning: Active learning allows for selection of the training data actively and with extra flexibility. This reduces the need for a large amount of labeled data by influencing the selection of data required for training [24]. The primary motivation of active learning starts from the cost and time of collecting labeled training dataset.

5) Reinforcement Learning: Reinforcement learning exposes and interacts with its environment and learns from the consequences of its action using trial and error. It is trained to make accurate decisions for the future action by capturing the learned knowledge and its experience [22].

6) Ensemble Learning: Ensemble learning combines multiple weak classifiers to create a stronger classifier model [25] by taking their individual decisions and their predictions to combine them. Boosting and Bagging are examples of ensemble learning.

B. Machine Learning based on Depth

Schmidhuber et al. [26] classify machine learning into shallow and deep learning which distinguishes the machine learning techniques based on how deep the credit assignment path is.

1) Shallow Learning: Shallow learning refers to the approach of standard machine learning models which do not utilize multiple hidden connection or layers. Shallow learning models do not suffer from vanishing gradient and the complexity of computations that come from the growth of connections. However, shallow models are usually limited and unable to capture correlation across the modulates [27].

2) Deep Learning: Deep learning involves the use of a multi-layer stack of simple modules [28]. Deep Learning overcomes scalability and complicated problems and is mostly being used for solving major critical scientific related problems on a large scale [9].

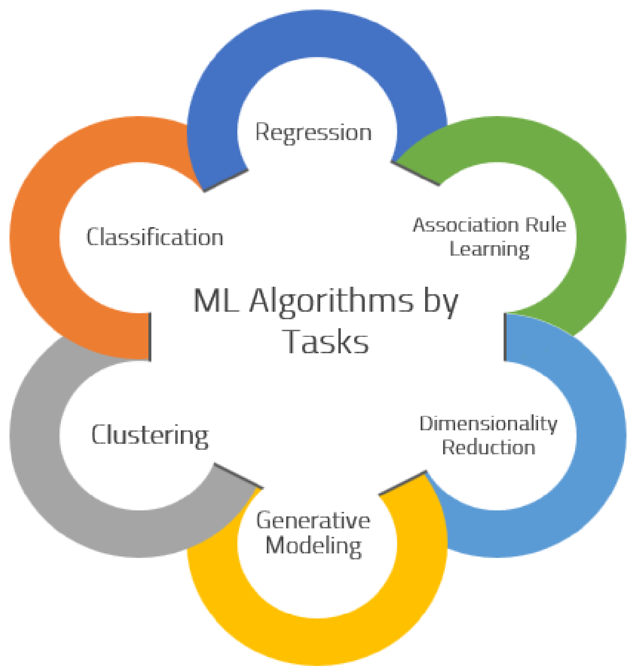

Fig. 2. Machine Learning Tasks

C. Machine Learning based on Tasks

Machine learning is used to perform various types of tasks based on the required approach and the type of data available. All machine learning techniques can be divided into six task categories as illustrated in Figure 2.

IV. Adversarial Machine Learning

Adversarial attacks have been studied for more than a decade now. However, the first notable discovery in adversarial attacks for computer vision was by Szegedy et al. [29] who reported that a small perturbation in the form of a carefully crafted input could confuse a deep neural network to misclassify an image object. Other researchers have demonstrated the use of adversarial attacks beyond image classification [30][31][32][33].

A. Adversarial Samples

A major component of an adversarial attack is the adversarial example. An adversarial sample consists of an input to a machine learning model which has been perturbed.

For a particular dataset with features x and label y, a corresponding adversarial sample is a specific data point x' which causes a classifier c to predict a different label on x' other than y, but x' is almost indistinguishable from x.

The adversarial samples are created using one of many optimization methods known as adversarial attack methods. Crafting adversarial samples involves solving an optimization problem to determine the minimum perturbation which maximizes the loss for the neural network

Considering an input x, and a classifier f, the optimization goal for the adversary is to compute such perturbation with a small norm, measured w.r.t some distance metric, that would modify the output of the classifier such that

$$f(x + \delta) \neq f(x)$$

where δ is the perturbation.

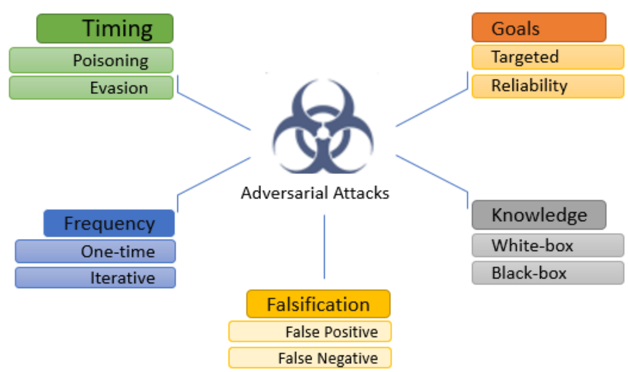

Fig. 3. Adversarial Threat Model

B. Adversarial Threat Model

We examine the threat model in Figure 3 to consider the goals and capabilities of any adversary for a machine learning system. Adversarial attack threats may be considered based on the attacker's knowledge, attack goals, attack timing, attack frequency, and attack falsification.

- *Attacker's Knowledge - White-box vs Black-box attacks*: In white-box attacks, the attacker knows the exact information about the learning algorithm or the learned model. In contrast, black-box attacks assume that the adversary has limited or no knowledge about the learning algorithm or the learned model.
- *Attacker's Goal - Targeted vs Reliability attacks*: In targeted attacks, the attacker has a specific goal with regard to the model decision. Most commonly, the attacker would aim to induce a definite prediction from the machine learning model. On the other hand, a reliability attack occurs when the attacker only seeks to maximize the prediction error of the machine learning model without necessarily inducing a specific outcome. Yevgeny et al. [17] have noted that the distinction between reliability and targeted attacks becomes blurred in attacks on binary classification tasks such as malware binary classification.
- *Attack Timing - Evasion vs Poisoning attacks:* In evasion attacks, also known as exploratory attack or attack at decision time, the attacker aims to confuse the decision of the machine learning model after it has been learned as shown in Figure 4. This is in contrast to poisoning attacks, also known as causative attack, which involves adversarial corruption of the training data before training to induce a wrong prediction from the machine learning mode as shown in Figure 5.

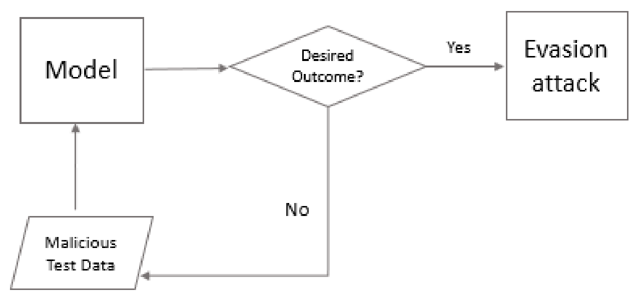

Fig. 4. Evasion Attack

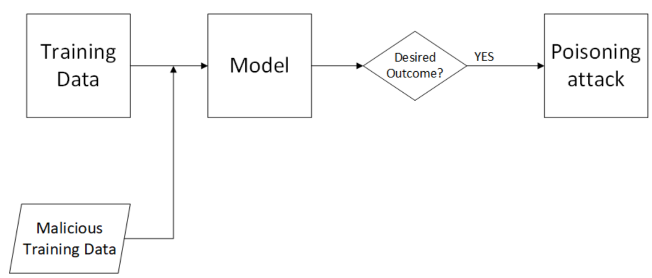

Fig. 5. Poisoning Attack

- *Attack Frequency - One-shot vs Iterative attacks:* Adversarial attacks are also classified based on the frequency with which the adversarial samples are updated or optimized. One-shot or one-time attacks are attacks in which the adversarial examples are optimized just once. Iterative attacks, however, involve updating the adversarial examples multiple times. By updating the adversarial samples multiple times, the samples are better optimized and perform better compared to one-shot attacks. However, iterative attacks cost more computational time to generate. Adversarial attacks against certain machine learning styles which are computationally intensive such as reinforcement learning usually demand one-shot attacks as the only feasible approach [34]

- *Attack Falsification - False-positive vs False-Negative attacks:* False positive attacks cause a machine learning model to mis-classify a negative sample as a positive one. For example, a malware is being classified as a benign in a false positive attack. A false negative attack in contrast results in a positive sample being misclassified as a negative sample.

V. Machine Learning Tasks in Network Security

A. *Regression (Prediction) Tasks in Network Security*

Regression tasks involve methods for predicting next series of information from prior data [20]. In network security, a regression model is used to predict relevant parameters from network packet and then draw a comparison between them with the regular parameters [35]. Kolosnjaji et al. [36] used regression models for predicting system calls for executable processes to derive a relationship the actual processes. Also, regression algorithms are used for anomaly detection in computer networks, user behavior analytics (e.g., Human Interaction Proofs) and predicting anomalies in process behaviour [37] such as credit card fraud transactions. Example of ML algorithms for regression tasks are Linear regression, Polynomial regression, Ridge regression, Support Vector Regression, Decision trees, Random Forest. Deep Learning algorithms for regression tasks include Artificial neural networks, Recurrent neural networks, Neural tuning machines and Differentiable Neural Computer.

B. *Classification Tasks in Network Security*

Classification tasks involve categorizing data into different categories from pre-labeled examples [38]. In network security, classification task model is used in detecting known types of fraud [37], and for grouping different users like in social spammers [39]. Also, it is used to categorize programs and files as malware, spyware, and ransomware, and to identify different classes of network attacks. Example of algorithms for classification tasks are Logistic regression, Decision Trees, Random Forests, Artificial neural networks, Support vector machines and Convolutional Neural Networks.

C. *Clustering Task in Network Security*

Clustering tasks are used to group the input data by similarity or patterns into unknown classes [22]. Clustering task is used to compare industry and business processes [40] and detect outliers. Girma et al. used clustering to detect DDOS attacks [41]. Also, the clustering task is used in forensic analysis. Examples of clustering algorithms are K-nearest neighbors, K-means, Mixture model, Self-organized Maps (SOM) and Kohonen Networks.

D. *Association Rule Learning (Recommendation) Task in Network Security*

Association Rule Learning (ARL) involves the discovery of rules and relations that describe large portions of data and find the link between X and Y where the X is the antecedent and Y is the consequence of rule [22]. Common ARL algorithms include Apriori, Euclat, and Deep belief networks.

E. *Dimensional Reduction (Generalization) Task In Network Security*

Dimensionality reduction encodes a multi-dimensional dataset into a compact lower dimensional representation while preserving as much information as possible in the original dataset. Example algorithms are Principal Component Analysis, Singular Value Decomposition, Linear Discriminant Analysis, Independent Component Analysis.

F. *Generative Modelling Task In Network Security*

Generative modelling tasks involve training a model by learning the data distribution within a training dataset. Subsequently, new data points are generated and associated decisions are made to simulate an entirely new data sample. Examples of algorithms for generative modelling tasks include Markov Chains, Variational Auto-encoders, Generative Adversarial Networks (GANs) and Boltzmann Machines.

VI. Applications of Machine Learning in Network Security

Machine learning techniques have been increasingly used to carry out a wide range of tasks in network security [42]. In this section, we review and highlight some applications of machine learning in network security by classifying them into five categories.

A. *Machine Learning for Network Protection*

Intrusion Detection Systems (IDS) are essential solutions for monitoring events dynamically in a computer network or system. Essentially there are two types of IDS (signature based and anomaly based) [43]. Signature based IDS detects attacks based on the repository of attacks signatures with no false alarm [44]. However, zero-day attacks can easily bypass signature-based IDS. Anomaly IDS [44] uses machine learning and can detect a new type of attacks and anomalies. A typical disadvantage of anomaly IDS is the tendency to generate a significant number of false positive alarms.

1) Hybrid Approach for Alarm Verification: Sima et al. [45] designed and built Hybrid Alarm Verification System that requires processing a significant number of real-time alarms, high accuracy in classifying false alarms, perform historical data analysis. The proposed system consists of three components: Machine Learning, Stream processing and Batch processing (Alarm History). Machine learning model trained offline and used for verification service that can immediately classify true or false alarms. They used different machine learning algorithms in the experiments to show the effectiveness of their system where the accuracy achieves more than 90% in a stream of 30K alarms per second [45].

2) Learning Intrusion Detection: Laskov et al. [46] worked in developing a framework to compare the supervised learning (classication) and unsupervised learning (clustering) techniques for detecting intrusions and malicious. They used different methods in supervised learning to evaluate the work include k-Nearest Neighbor (kNN), decision trees, Support Vector Machines (SVM) and Multi-Layer Perception (MLP). Also, k-means clustering was utilized, with single linkage clustering as unsupervised algorithms. The evaluation was ran under two scenarios to evaluate how much the IDS could generalize its knowledge to new malicious activities. The supervised algorithms showed better classification with the known attacks. The best result among the supervised algorithm was the decision tree algorithm whiched achieved 95% true positive and 1% false positive rate, followed by MLP, SVM and then KNN. If there were new attacks not previously seen in the training data, the accuracy decreases significantly. However, the unsupervised algorithms performed better for unseen attacks and did not show signicant dierence in accuracy for seen and unseen attacks [46].

B. Machine Learning for Endpoint Protection

Malware detection is a significant part of endpoint security including workstations, servers, cloud instances, and mobile devices. Malware detection is used to detect and identify malicious activities caused by malware. With the increase in the variety of malware activities, the need for automatic detection and classifier amplifies as well. The signature-based malware detection system is commonly used for existing malware that has a signature but it not suitable for unknown malware or zero-day malware. Machine learning can cope with this increase and discover underlying patterns in large-scale datasets [36].

1) Automatic Analysis of Malware Behavior: Rieck et al. [47] successfully proposed a framework for analyzing malware behavior automatically using various machine learning techniques. The framework allows clustering similar malware behaviors into classes and assigns new malware to these discovered classes. They designed an incremental approach for the behavior analysis that can process various malware behaviors and reduce the run-time defense against malware development comparing to other analysis methods and provide accurate discovery of novel malware. To implement this automatic framework, they collected a large number of malware samples and monitored their behaviors using a sandbox environment and learn those behaviors using Clustering and Classification algorithms [47].

2) Automated Multi-level Malware Detection System: In [48], authors proposed Advanced Virtual Machine Monitor-based guest-assisted Automated Multilevel Malware Detection System (AMMDS) that affect both Virtual Machine Introspection (VMI) and Memory Forensic Analysis (MFA) techniques to mitigate in real time symptoms of stealthily hidden processes on guest OS [48]. They use different machine learning techniques such as Logistic Regression, Random Forest, Naive Bayes, Random Tree, Sequential Minimal Optimization (SMO), and J48 to evaluate the AMMDS and the results achieve 100%.

3) Classication of Malware System Call Sequences: Kolosnjaji et al. [36] focused on the utilization of neural networks by stacking layers according to deep learning to improve the classication of newly retrieved malware samples into a predened set of malware classes. They constructed Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) layers for modeling System Call Sequences. The sequences used by the CNN layers was based on a set of n-grams. The presence of the n-grams and their relation were counted in a behavioral trace. The RNN on the other hand used sequential information to train the model. A dependence between the system call appearance and the system call sequence was however maintained. If this model was trained properly, it usually provided better accuracy on subsequent data and most often captured more training set information. This deep learning technique for capturing the relation between the n-grams in the system call sequences was deemed to be relatively efficient as it achieved 90% average accuracy, precision and recall for most of the malware families [36].

4) A Hybrid Malicious Code Detection Method: Li et al. [49] proposed a hybrid malicious code detection scheme based on AutoEncoder and Deep Belief Networks (DBN). They used the AutoEncoder to reduce the dimensionality of data by extracting the main features. Then they used the DBN that composed multilayer Restricted Boltzmann Machines (RBM) and a layer of BP neural network to detect malicious code. The BP neural network has an input vector from the last layer of RBM based on unsupervised learning and then use supervised learning in the BP neural network. They achieved the Optimal hybrid model. The experiment results that are verified by KDDCUP'99 dataset show higher accuracy compared to a single DBN and reduce the time complexity [49].

C. Machine Learning for Application Security

Various machine learning tasks used for application security including malicious web attack detection, phishing detection and spam detection.

1) Detection of Phishing Attacks: Basnet et al. [50] studied and compared the effectiveness of using different machine learning algorithms for classification of phishing emails using many novel input features that helps in detecting phishing attacks. The training dataset is labeled with phishing or legitimate email. They used unsupervised learning to extract features without prior training directly and provides fast and reliable knowledge from the dataset. They used 4000 emails in total, A total of 2000 emails used for testing. They used Support Vector Machines (SVM), Leave One Model Out, Biased SVM, Neural Networks, Self Organizing Maps (SOMs) and K-Means on the dataset. Consistently, Support Vector Machine achieved the best results. The Biased Support Vector Machine (BSVM) and NN have an accuracy of 97.99% [50].

2) Adaptively Detecting Malicious Queries in Web Attacks: Don et al. [51] proposed a new system called AMODS and learning strategy called SVM HYBRID for detecting web attacks. AMODS is an adaptive system that aims to periodically update the detection model to detect the latest web attacks. The SVM HYBRID is an adaptive learning strategy which was implemented primarily for reducing manual work. The detection model was trained using dataset which was obtained from an academic institutes web server logs. The proposed detection model outperformed existing web attack detection methods with an FP rate of 0.09% and 94.79% F-value. The SVM Hybrid system obtained a total number of malicious queries equal to 2.78 times by the popular SVM method. Also, the Web Application Firewall (WAF) can use malicious queries to update the signature library. The significant queries were used for updating the detection model which consisted of a meta-classifier as well as other three base classifiers [51].

3) URLNet -Learning a URL Representation with Deep Learning for Malicious URL Detection: Le et al. [52] proposed an end-to-end deep learning framework which did not require sophisticated feature. URLNet was introduced to address several limitations which was found with the other model approaches. This framework learns from the URL directly how to perform a nonlinear URL embedding which then enabled it to successfully detect various Malicious URLs. Convolutional Neural Networks (CNN) were applied to both the characters and words of each URL to discover the URL embedding method. They also proposed advanced word-embedding techniques to deal with uncommon words, which was a limitation being experienced by other malicious URL detection systems. The framework then learns from unknown works at testing phase [52].

D. Machine Learning for User Behavior Analytic

User behavior analytics is a cybersecurity process which involves analyzing patterns in human behaviors and detecting anomalies that give an indication of fraudulent activities or insider threats. Machine learning algorithms are used to detect such anomalies in user actions such as unusual login tries and to infer useful knowledge from those patterns.

1) Authentication with Keystroke Dynamics: Revett et al. [53] proposed a system using Probabilistic Neural Network (PNN) for keystroke dynamics that captures the typing style of a user. A system comprising of 50 user login credential keystrokes was evaluated. The authors [53] used eight attributes to monitor the enrollment and authentication attempts. An accuracy of 90% was obtained in classifying legitimate users from imposters. A comparison of the training time between the PNN system and a Multi-Layer Perception Neural Network (MLPNN) showed that the PNN was four times faster.

2) Text-based CAPTCHA Strengths and Weaknesses: Bursztein et al. [54] in a study showed that several well known websites still implemented technologies that have been proven to be vulnerable to cyber attacks. In the study, an automated Decaptcha tool was tested on numerous websites including well known names such as eBay, Google and Wikipedia. It was observed that 13 out of 15 widely used web technologies were vulnerable to their automated attack. They had a significant success rate for most of the websites. Only Google and Recaptacha were able to resist to the automated attack. Their study revealed the need for more robust CAPTCHA designs in most of the widely used schemes. Authors recommended that the schemes should not rely on segmentation alone because it did not provide sufficient defense against automated attacks.

3) Social Network Spam Detection: K. Lee et al. [39] proposed social network spam detection that gathers legitimate and spam profiles and feeds them to Support Vector Machine (SVM) model. K. Lee et al.[39] selected two social networks: Twitter and MySpace to evaluate the proposed machine learning system. They collected data over months and feed them to the SVM classifier. The dataset contains 388 legitimate profiles and 627 spam profiles collected from MySpace, and 104 legitimate profiles and 168 profiles between promoters and spammers collected from Twitter. The system achieved a low false positive rate and high precision up to 70% for MySpace and 82% for Twitter.

E. Machine Learning for Process Behavior Analytic

Machine learning applications usually necessitate the need to learn and have some domain knowledge about business process behaviors in order to detect anomalous behaviors. Machine learning could be used for determining fraudulent transactions within banking systems. Also it was been successfully used for identifying outliers, classifying types of fraud and for clustering various business processes.

1) Anomaly detection in Industrial Control Systems: Kravch et al.[40] performed a successful study on Secure-Water Treatment Testeb (SWat) using Deep Convolutional Neural Networks CNN to detect most of attacks on Industrial Control System (ICS) with a low false positive. The anomaly detection method was based on the statistical deviation measurement of the predicted value. They performed the study using 36 different attacks from SWat. The authors in [40] proofed that using 1D convolutional networks in anomaly detection in ICS outperformed the recurrent networks.

2) Detecting Credit Card Fraud: Traditionally, the Fraud Detection System uses old transactions data to predict a new transaction. Fraud Detection System (FDS) should encounter various potential challenges and difficulties to achieve high accuracy and performance [55]. The traditional detection method does not solve all problems and challenges including imbalanced data where there is a small chance of transactions are fraudulent. Wrong classification and overlapping data and Fraud detection cost are other major challenges [55]. Chen et al. [37] proposed an approach to solving the listed challenges and problems for Credit Card fraud. They introduced a system to prevent fraud from the initial use of credit cards by collecting user data from online questionnaire based on consumer behavior surveys. They used various classifiers models: decision tree (C5.0, CandRT, CHAID) and SVM (linear and radial basis, Kernels of polynomial, sigmoid). They use three datasets to develop questionnaire-responded transaction (QRT) model to predict new transaction.

3) Deep Learning Techniques for Side-Channel Analysis: Prouff et al. [56] defined Side-Channel Analysis as a type of attack that attempts to leak information from a system by exploiting some parameters from the physical environment [56]. This attack was utilizing the running-time of some cryptographic computation, especially in the block ciphers. The capability of a system to resist side-channel attacks (SCA) requires an evaluation strategy that focuses on deducing the relationship between the device behavior and the sensitivity of the information that is common in classical cryptography. The authors in [56] focused on proposing an extensive study of using deep learning algorithms in the Side-Channel Analysis. Also, they focused on the hyper-parameters selection to help in designing new deep learning classifier and models. They confirmed that the Convolutional Neural Networks (CNN) models are better in detecting SCA. Their proposal system outperformed the other tested models on highly desynchronized traces and had the best performance as well on small desynchronized trace [56].

VII. Adversarial Attack methods and Algorithms

We recall that adversarial attacks could be deployed either during decision time (evasion attacks) or during training time (poisoning attacks). In each case, the training algorithm (for poisoning attacks) or the learned model (for evasion attacks) is being manipulated with some form of carefully crafted input known as the adversarial samples. A common trend among the attack methods below reveals that the robustness of a machine learning model to a large extent depends on the ability of an attacker to find an adversarial sample that is as close as possible to the original input. In this section, we evaluate the primary methods for generating adversarial samples. It should be noted that recent research has shown the limitations of some earlier methods that are still listed here for reference even though more effective methods have been introduced.

1) L-BFGS: Szegedy et al. [29] studied how adversarial examples could be generated against neural networks for image classification. The L-BFGS (Limited Broyden-Fletcher-Goldfarb-Shanno) method was then introduced which used an expensive linear search method to find the optimal values of the adversarial samples.

2) Fast Gradient Sign Method (FGSM): In a different approach proposed by Goodfellow et al. [1], adversarial examples are created by finding the maximal direction of positive change in the loss. This is a faster method compared to the L-BFGS method since only a one-step gradient update is performed along the direction of the sign gradient at each level.

3) Basic Iterative Method (BIM): A major limitation of the Fast Gradient Sign Method and similar attack methods is that they work based on the assumption that the adversarial samples can be fed directly into the machine learning model. This is far from being practical since most attackers would seek to access the machine learning models through devices such as sensors [57]. The Basic Iterative Method proposed in [58] overcomes this limitation by running the gradient update in multiple iterations.

4) Jacobian-based Saliency Map Attack (JSMA): The Jacobian-based Saliency Map Attack (JSMA) was introduced by Papernot et al. [4]. For the attack, the Jacobian matrix of a given sample is computed to find the input features of that sample which most significantly impacts the output. Subsequently, a small perturbation is created based on that input feature for generating the adversarial attack.

5) DeepFool: DeepFool was proposed by Moosavi et al. [3] as a method for creating adversarial examples by finding out the closest distance between original input and the decision boundary for adversarial examples. They were able to determine that by using a related classifier, the closest distance which would correspond to the minimal perturbation for creating an adversarial sample will be the distance to the hyperplane of the related classifier.

6) Carlini and Wagner Attack: Carlini et al. [59] developed a targeted attack specifically for existing adversarial defense methods. It was discovered that defenses such as defensive distillation were ineffective towards the Carlini and Wagner attack.

VIII. Adversarial Attack Classification

In this section, we introduce a classification method for adversarial attacks in network security. We base our classification on a taxonomy of network security applications which was earlier discussed in section VI.

A. Adversarial attacks on ML for endpoint protection

A major component of endpoint protection in network security is malware detection. Yet, malware detection remains a challenging problem in network security. Between 2009 and 2019, the number of new malware digital signatures has increased by over 2000 percent [60]. Therefore, traditional malware detection systems that rely solely on digital signatures have become less effective. Significant effort has been made in the use of machine learning to protect against malware attacks. Several researches have shown the vulnerability of these machine learning models to adversarial attacks.

1) Iagodroid: One of the earliest attacks against machine learning based malware detection systems was the Iagodroid attack [61]. Iagodroid uses a method to induce mislabelling of malware families during the triaging process of malware samples.

2) Texture Perturbation Attacks: Researchers have deployed visualization techniques similar to computer vision and adapted it for malware classification [62]. This involves conversion of malware binary code into image data. The Adversarial Texture Malware Perturbation Attack (ATMPA) achieved a 100 percent effectiveness in defeating visualization based machine learning malware detection system and also resulted in 88.7 percent transfer-ability rate [19]. The attack model for ATMPA works by allowing the attacker to distort the malware image data during the visualization process.

3) EvnAttack: EvnAttack is an evasion attack model that was proposed in [32] which manipulates an optimal portion of the features of a malware executable file in a bi-directional way such that the malware is able to evade detection from a machine learning model based on the observation that the API calls differently contribute to the classification of malware and benign files.

4) AdvAttack: AdvAttack was proposed in [30] as a novel attack method to evade detection with the adversarial cost as low as possible. This is achieved by manipulating the API calls by injecting more of those features which are most relevant to benign files and removing those features with higher relevance scores to malware.

5) MalGAN: To combat the limitations of traditional gradient-based adversarial example generation, the use of a generative adversarial network (GAN) based algorithm for generating adversarial examples has been proposed. Generative models have been mostly used for input reconstruction by encoding an original image into a lower-dimensional latent representation [2]. The latent representation of the original input can be used to distort the initial input to create an adversarial sample. MalGAN proposed by [63] leverages on generative modeling techniques to evade black-box malware detection systems with a detection rate close to zero.

6) Slack Attacks: A byte-based convolutional neural network (MalConv) was introduced in [64]. Unlike image perturbation attacks [29], where the fidelity of the image is of little concern, attacks that alter the binaries of malware files must maintain the semantic fidelity of the original file because altering the bytes of the malware arbitrarily could affect the malicious effect of the malware. This problem could be solved by appending adversarial noise to the end of the binary [33]. This prevents the added noise from affecting the malware functionality. The Random Append attack and Gradient Append attacks are two types of append attacks which work by appending byte values from a uniform distribution sample and gradually modifying the appended byte values using the input gradient value. Two additional variations of append attacks; the benign append and the FGM Append were introduced by [65] which improves the long convergence time experienced in previous attacks.

When malware binaries have exceeded the model's maximum size, it is impossible to append additional bytes to them. Hence a slack attack proposed by [65] exploits the existing bytes of the malware binaries. The most common form of the slack attack is the Slack FGM Attack which defines a set of slack bytes that can be freely modified without breaking the malware functionality.

B. Adversarial attacks on ML for network protection

1) IDSGAN: IDSGAN was proposed in [66] for generating adversarial attacks targeted towards intrusion detection systems. IDSGAN is based on the Wasserstein GAN [67] which uses a generator, discriminator and a black-box. The discriminator is used to imitate the black-box intrusion detection system and at the same time provide the malicious traffic samples.

2) TCP Obfuscation Techniques: Another method for evading machine learning based intrusion detection systems is the use of obfuscation techniques.[68] proposed the modification of various properties of network connections to obfuscate a TCP communication which successfully evades a wide variety of intrusion detection classifiers.

C. Adversarial attacks on ML for Application Security

1) Attacks on Statistical Spam filters: Several spam filters such as SpamAssasin, SpamBayes, Bogofilter are based on the popular Naive Bayes Machine learning algorithm which was first applied to filtering junk email in 1998 [69]. A variety of good word attacks introduced by [70] were successfully evading the machine learning models from detecting spam or junk emails.

D. Adversarial attacks on ML for User Behavior Analytics

1) Attacks against crowd-turfing detection systems: Machine learning techniques are used to identify misbehavior includes fake users in social networks and detect users who pays for sites to have fake accounts. Malicious crowdsourcing or crowd-turfing systems are used to connect users who are willing to pay, with workers who carry out malicious activities such as generation and distribution of fake news, or malicious political campaigns. Machine learning models have been used to detect crowdturfing activity with up to 95 percent accuracy particularly in detecting the accounts of crowdturfing workers [71]. However, malicious crowdsourcing detection systems are highly vulnerable to adversarial evasion and poisoning attacks.

2) Attacks Against ML for Keystroke Dynamics: Authors [72] created adversarial keystroke samples that misled an otherwise accurate classifier into accepting the artificially generated keystroke samples as belonging to an authentic user.

E. Adversarial attacks on ML for Process Behavior Analytics

1) Attacks against ML for credit card fraud detection: [73] examined how a logistic regression classifier used as a fraud detection mechanism, could be adversarially attacked to cause a number of fraudulent transactions to go undetected. Previous studies have similar models which are based on game theory to investigate adversarial attacks against credit card fraud detection and email spam detectors. However, the authors introduced a new framework which successfully produced an improved AUC score on multiple iterations of the validation sets compared to the performance of the models which credit card companies had previously used.

IX. Evaluating Adversarial Risk

In discussing adversarial risk, we introduce the concept of discriminative and directive autonomy of machine learning models. The two-fold goal of an adversarial risk model is to evaluate the likelihood of success of an adversarial attack against a machine learning model, and the consequence of that attack if successful. We present in this paper, an adversarial risk model shown in Figure 6 based on the level of autonomy of the machine learning model with respect to the learning style and task. The concept of discriminative autonomy and directive autonomy of the machine learning

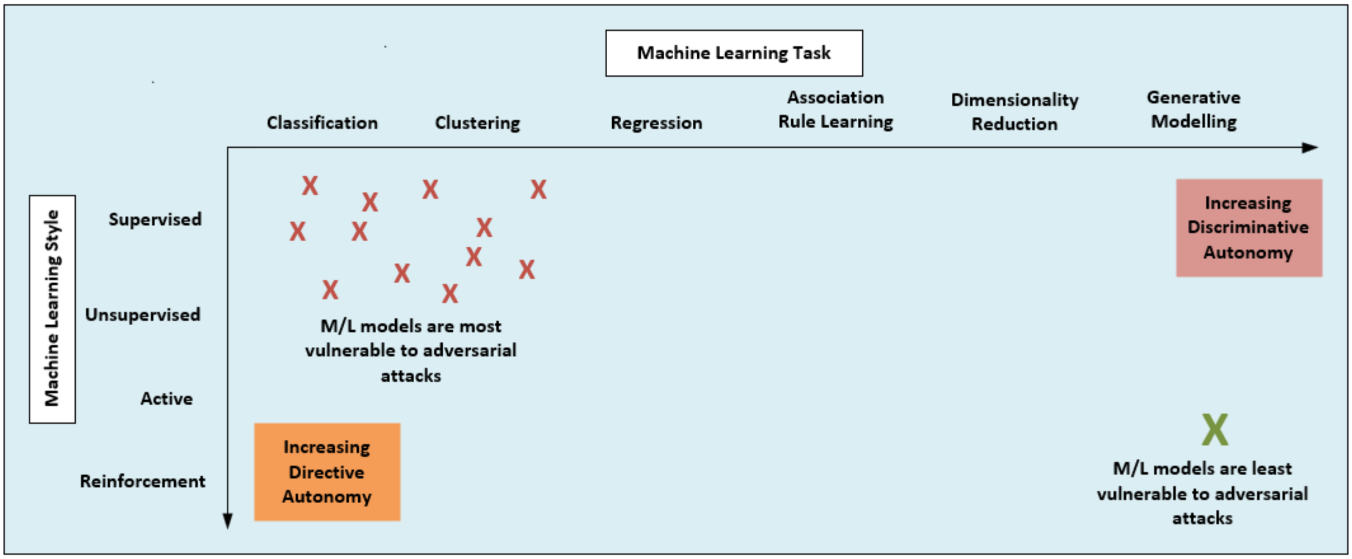

Fig. 6. Adversarial Risk Model

models represents a novel approach for evaluating the relative adversarial risk of a machine learning model.

- *Discriminative Autonomy:* The discriminative autonomy is directly related to the type of task being performed by the machine learning model. Machine learning tasks such as classification are highly dependent on the input data. As such, they have lower discriminative or conditional autonomy compared to tasks such as generative modeling which depend less on the input data when predicting an outcome.
- *Directive autonomy:* The directive autonomy of a machine learning model is a function of the machine learning style. In supervised machine learning, there is less directive autonomy since the model needs to be first learned with some form of labeled data. Machine learning styles such as reinforcement learning depend less on a model being learned with any form of training data and posses much higher directive autonomy.

X. Defending Against Adversarial Attacks

Barreno et al. [15] first proposed three broad approaches for defending machine learning algorithms against adversarial attacks. Regularization, Randomization, and Information hiding. Yuan et al. [57] classified the defenses into two broad strategies. Proactive strategies and reactive strategies. In this section, we provide the most common attack methods in use today and classify them based on the strategy and approach. Our classification is illustraded in Table II.

1) Gradient masking: The gradient masking method modifies a machine learning model in an attempt to obscure its gradient from an attacker. Nayebi et al [74] demonstrated the effect of gradient masking by saturating the sigmoid network which results in a vanishing gradient effect.

2) Defensive Distillation: Distillation technique was originally proposed by Hinton et al. [78] for transferring knowledge from large neural networks to smaller ones. It was adapted by Papernot et al. [75] to defend against adversarial crafting by using the output of the original neural network to train a smaller network rather than using the distillation as originally proposed by Hinton. Defensive distillation was initially tested against adversarial attacks in computer vision, but further research is required to determine its effectiveness in other applications such as malware detection.

TABLE II
Classification of Defense Methods

Method	Strategy	Approach
Gradient Masking [74]	Proactive	Information hiding
Defensive Distillation [75]	Reactive	Randomization
Adversarial Training [29]	Proactive	Regularization
Detecting adversarial Examples [76]	Reactive	Regularization
Feature Reduction [77]	Proactive	Regularization

3) Adversarial Training: Szegedy et al. [29] originally proposed a three-step method known as adversarial training for defending against adversarial attacks. 1, Train the classifier on the original dataset 2, Generate adversarial samples 3, Iterate additional training epochs using the adversarial examples. Adversarial training improves the classification performance of the machine learning model and makes it more resilient to adversarial crafting

4) Detecting Adversarial Examples: Several approaches are used to detect the presence of adversarial examples in the training phase of a machine learning model. One of such approaches proposed by [76] works on the premise that adversarial examples have a higher uncertainty that clean data and uses a Bayesian neural network to estimate the extent of uncertainty in the input data to detect the adversarial samples. Other approaches include the use of probability divergence proposed by [79] as well as the use of an auxiliary network of the original network introduced by Metzen et al.

in [80]

5) Feature Reduction: Other potential defenses for adversarial attacks have been proposed. Simple feature reduction was evaluated by Grosse et al. [77] but was found inadequate in defending against adversarial attacks.

6) Ensemble Defenses: Similar to the idea of ensemble learning which combines one or more machine learning techniques, researchers have also proposed the use of multiple defense strategies as a defense technique against adversarial examples. PixelDefend was proposed by [81] to combine adversarial detecting techniques with one or more other methods for creating a more robust defense against adversarial attacks.

XI. Defenses against adversarial attacks in network security

In this section, we introduce specific solutions that have been proposed for defending against adversarial attacks in network security.

1) KUAFUDET Camouflage Detector: One method to improve the accuracy of machine learning based malware classifiers is a novel malware camouflage detector - KUAFUDET which significantly reduces false negatives and boosts the detection accuracy by at least 15 percent. KUAFUDET utilizes two-phase learning enhancing method which learns the features of a malware sample through adversarial detection [82].

2) SecureDroid: Another approach for adversarial defense against machine learning based malware detection is the SecureDroid defense [61] which integrates two methods - the SecCLS and SecENS methods for enhancing Android malware detection.

3) SecDefender: To achieve resilience against evasion attacks, SecDefender [32] was proposed as a secure learning paradigm for malware detection which is based on classifier retraining techniques.

4) DroidEye: Adversarial android malware attacks can be prevented with a system called DroidEye which implements count featurization for feature transformation to harden the machine learning classifier against the attacks [31].

5) SecureMD: In SecureMD proposed in [30], the machine learning malware detection classifier is enhanced through the use of security regularization terms utilizing a fitting constraint and a smoothness constraint. SecureMD has improved detection accuracy by up to 93 percent.

6) Weighted Bagging: Biggio et al. [83] proposed the use of bagging classifiers for preventing poisoning attacks against machine learning based network protection systems. Bagging uses bootstrap aggregation techniques to create bootstrap replicates of the training set. The classifier is then trained on the bootstrap replicates, and the predictions are aggregated.

7) Reject on Negative Impact (RONI): The reject on negative impact technique has been proven to achieve 100 percent accuracy in detecting adversarial attacks against intrusion detection systems.

8) Deepcloak: DeepCloak introduced by Ji Gao et al. [84] works by removing unnecessary features that may be used for generating adversarial samples.

XII. Feature Space and Problem Space - Dimensional Space Classification

In this section, we categorize adversarial attacks based on feature space and problem space. In the field of machine learning, a problem space also known as state space can be defined as a dimensional representation of all the possible configurations of the objects in a problem determination context. Conversely, a feature space is defined as the n dimensional space in which all variables in the input dataset are represented. We take as an example an intrusion detection dataset with 70 variables, this represents a 70-dimensional feature space.

A feature space adversarial attack in the context above will seek to alter the feature space by making changes within the 70-dimensional feature space. A feature space attack modifies the features in the instance directly while a problem space adversarial attack modifies the actual instance itself. Using an example of malware adversarial attacks, a feature space adversarial malware attack will only modify the feature vectors but no new malware is created. A problem space adversarial malware attack will modify the actual instance from the source to produce a new instance of the malware. Compared to a problem space adversarial attack, a feature space adversarial attack does not generate a new sample but creates a new feature vector.

Feature space modelling of an adversarial sample is a method in which an optimization algorithm is used to find the ideal value out of a finite number of arbitrary changes made to the features. In a feature space adversarial attack, the attacker's objective is to remain benign without generating a new instance.

Feature space and problem space dimensional classification of various adversarial attacks are shown in Tables III and IV respectively. From our observation, adversarial attacks in problem space are more difficult to generate and also more difficult to defend against.

XIII. Summary and evaluation

We observed an increased risk of adversarial vulnerability of machine learning models in network security with reduced discriminative autonomy and directive autonomy. Similarly, we observed a reduced risk of adversarial vulnerability with increased discriminative autonomy and directive autonomy. As illustrated in the adversarial risk model shown in Fig. 6, the discriminative autonomy directly relates to the machine learning tasks while the directive autonomy relates to the machine learning style. The reason for the adversarial sensitivity of the machine learning models to the discriminative and directive autonomy based risk model is still an area of open research.

Previous approaches on making machine learning in network security more secure have advocated the development of machine learning models that are resilient to adversarial attacks. In this survey, we introduced the concept of an element of reduced risk of adversarial attacks based on an adversarial risk model. Our findings suggest that the adversarial risk model provides a promising future for the

TABLE III

SUMMARY OF FEATURE SPACE ADVERSARIAL ATTACKS IN NETWORK SECURITY AND THEIR EFFECTIVENESS

[* The misclassification rates provided in the table are as reported by individual authors in the respective publications. Since the platforms for experimentation were not standardized, these numbers are not considered as an ideal benchmark comparison of adversarial misclassification effectiveness]

Attacks	Classification	Timing	Goal	Information	Falsification	Frequency	*Misclassification
ATMPA [62]	Endpoint protection	Poisoning	Targeted	White-box	False Positive	Iterative	97%
Good word Attacks [70]	Application Security	Evasion	Targeted	White-box	False Positive	Iterative	40%
Slack Attack [65]	Endpoint protection	Evasion	Targeted	White-box	False Positive	One-Shot	20%
Attack on Keystroke dynamics [72]	User Behavior	Evasion	Targeted	White-box	False Positive	Iterative	50%
IogoDroid [61]	Endpoint protection	Evasion	Targeted	White-box	False Positive	Iterative	97%
TCP Obfuscation Attack [68]	Network protection	Evasion	Targeted	White-box	False Positive	Iterative	70%
Attack on Crowdturfing detector [71]	User Behavior	Poisoning	Reliability	White-box	False Positive	Iterative	85%
Attack on Credit Card Fraud Detection [73]	Process Behavior	Poisoning	Targeted	Black-box	False Positive	Iterative	NS
PDF Malware Evader [85]	Endpoint protection	Evasion	Targeted	White-box	False Positive	Iterative	100%
EvnAttack [32]	Endpoint protection	Evasion	Targeted	White-box	False Positive	One-shot	97%
AdvAttack [30]	Endpoint protection	Poisoning	Targeted	White-box	False Positive	Iterative	90%
Malware Detection Feature Selection [86]	Endpoint Protection	Poisoning	Reliability	Black-box	False Positive	Iterative	20%

TABLE IV

SUMMARY OF PROBLEM SPACE ADVERSARIAL ATTACKS IN NETWORK SECURITY AND THEIR EFFECTIVENESS

[* The misclassification rates provided in the table are as reported by individual authors in the respective publications. Since the platforms for experimentation were not standardized, these numbers are not considered as an ideal benchmark comparison of adversarial misclassification effectiveness]

Attacks	Classification	Timing	Goal	Information	Falsification	Frequency	*Misclassification
DeepDGA Attack [87]	Application Security	Poisoning	Targeted	Black-box	False Positive	One-Shot	80%
MalGAN Attack [63]	Endpoint protection	Evasion	Targeted	Black-box	False Positive	Iterative	90%
IDSGAN Atack [66]	Network protection	Evasion	Reliability	White-box	False Positive	Iterative	99%

security of artificial intelligence and machine learning in network security. Machine learning based network security applications that are more resilient to adversarial attacks can be designed by leveraging on the adversarial risk model.

We observed that the misclassification achieved by an adversarial attack is dependent significantly on the design of the adversarial attack algorithm with the context of each specific attack . White-box, Evasion attacks against endpoint protection systems (malware detection) are the most common attacks. While there is limited research in adversarial attacks against process behavior and user behavior analysis, use cases of machine learning in network security, endpoint protection, network protection and application security have been well researched.

We reviewed defenses against adversarial attacks on machine learning applications in network security. We note that there are two major limitations in the existing research on adversarial defenses. Firstly, most defenses are designed to protect against attacks on machine learning applications in computer vision. Secondly, the defenses studied are usually designed for a specific attack or a part of the attack. A generalized defense model against adversarial attacks is at best still theoretical as research on generalized defense models is in early stages [88]. Furthermore, our findings indicate that defenses against adversarial attacks are specific to a particular type of attack and are not necessarily transferable. Recent research [89] have studied the transferability in malware machine learning models in machine learning applications such as malware detection.

XIV. CONCLUSION

We present a first of its kind survey on adversarial attacks on machine learning in network security. The previous survey [9] that we reviewed had only discussed adversarial attacks against deep learning in computer vision.

We introduced a new classification for adversarial attacks based on applications of machine learning in network security and developed a matrix to correlate the various types of adversarial attacks with a taxonomy-based classification to determine their effectiveness in causing a misclassification. We also presented a novel idea of the concept of an adversarial risk model for machine learning in network security.

In our review on defenses against adversarial attacks, although there were numerous proposed defenses against specific adversarial attacks, research on generalized defenses against adversarial attacks is still not well established [88]. In our future work, we would study generalized defenses against adversarial attacks to understand if a generalized approach towards adversarial defenses will be effectively attainable. In addition, we would examine the interpretability of the adversarial risk model to further understand why the reduced adversarial vulnerability occurs, and its implications for other applications of machine learning such as computer vision and natural language processing.

ACKNOWLEDGEMENT

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the NSERC Discovery Grant program.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples (2014)," *arXiv preprint arXiv:1412.6572*.

[2] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 36–42, IEEE, 2018.

[3] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, 2016.

[4] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pp. 372–387, IEEE, 2016.

[5] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[6] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware c&c detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 59, 2016.

[7] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE access*, vol. 6, pp. 12103–12117, 2018.

[8] V. Duddu, "A survey of adversarial machine learning in cyber warfare," *Defence Science Journal*, vol. 68, no. 4, pp. 356–366, 2018.

[9] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *arXiv preprint arXiv:1801.00553*, 2018.

[10] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[11] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, 2019.

[12] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," *Applied Sciences*, vol. 9, no. 5, p. 909, 2019.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[14] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519, ACM, 2017.

[15] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pp. 16–25, ACM, 2006.

[16] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[17] Y. Vorobeychik and M. Kantarcioglu, "Adversarial machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–169, 2018.

[18] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018.

[19] X. Liu, Y. Lin, H. Li, and J. Zhang, "Adversarial examples: Attacks on machine learning-based malware visualization detection methods," *arXiv preprint arXiv:1808.01546*, 2018.

[20] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear estimation and classification*, pp. 149–171, Springer, 2003.

[21] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.

[22] E. Alpaydin, *Introduction to machine learning*. MIT press, 2009.

[23] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, no. 3, p. 4, 2006.

[24] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *ICML*, pp. 839–846, Citeseer, 2000.

[25] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000.

[26] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[27] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.

[28] L. Deng, D. Yu, *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[29] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[30] L. Chen and Y. Ye, "Secmd: Make machine learning more secure against adversarial malware attacks," in *Australasian Joint Conference on Artificial Intelligence*, pp. 76–89, Springer, 2017.

[31] L. Chen, S. Hou, Y. Ye, and S. Xu, "Droideye: Fortifying security of learning-based classifier against adversarial android malware attacks," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 782–789, IEEE, 2018.

[32] L. Chen, Y. Ye, and T. Bourlai, "Adversarial machine learning in malware detection: Arms race between evasion attack and defense," in *Intelligence and Security Informatics Conference (EISIC), 2017 European*, pp. 99–106, IEEE, 2017.

[33] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables," *arXiv preprint arXiv:1803.04173*, 2018.

[34] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2040–2042, International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[35] M. Zamani and M. Movahedi, "Machine learning techniques for intrusion detection," *arXiv preprint arXiv:1312.2177*, 2013.

[36] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Australasian Joint Conference on Artificial Intelligence*, pp. 137–149, Springer, 2016.

[37] Y. G. Şahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," 2011.

[38] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[39] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 435–442, ACM, 2010.

[40] M. Kravchik and A. Shabtai, "Anomaly detection; industrial control systems; convolutional neural networks," *arXiv preprint arXiv:1806.08110*, 2018.

[41] A. Girma, M. Garuba, and R. Goel, "Advanced machine language approach to detect ddos attack using dbscan clustering technology with entropy," in *Information Technology-New Generations*, pp. 125–131, Springer, 2018.

[42] V. Ford and A. Siraj, "Applications of machine learning in cyber security," in *Proceedings of the 27th International Conference on Computer Applications in Industry and Engineering*, 2014.

[43] J. Singh and M. J. Nene, "A survey on machine learning techniques for intrusion detection systems," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 11, pp. 4349–4355, 2013.

[44] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *Intelligent Systems and Informatics (SISY), 2017 IEEE 15th International Symposium on*, pp. 000277–000282, IEEE, 2017.

[45] A.-C. Sima, K. Stockinger, K. Affolter, M. Braschler, P. Monte, and L. Kaiser, "A hybrid approach for alarm verification using stream processing, machine learning and text analytics," in *International Conference on Extending Database Technology (EDBT), March 26-29, 2018*, ACM, 2018.

[46] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?," in *International Conference on Image Analysis and Processing*, pp. 50–57, Springer, 2005.

[47] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," *Journal of Computer Security*, vol. 19, no. 4, pp. 639–668, 2011.

[48] A. Kumara and C. Jaidhar, "Automated multi-level malware detection system based on reconstructed semantic view of executables using

machine learning techniques at vmm," *Future Generation Computer Systems*, vol. 79, pp. 431–446, 2018.

[49] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *methods*, vol. 9, no. 5, 2015.

[50] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach," in *Soft Computing Applications in Industry*, pp. 373–383, Springer, 2008.

[51] Y. Dong, Y. Zhang, H. Ma, Q. Wu, Q. Liu, K. Wang, and W. Wang, "An adaptive system for detecting malicious queries in web attacks," *Science China Information Sciences*, vol. 61, no. 3, p. 032114, 2018.

[52] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, "Urlnet: Learning a url representation with deep learning for malicious url detection," *arXiv preprint arXiv:1802.03162*, 2018.

[53] K. Revett, F. Gorunescu, M. Gorunescu, M. Ene, P. S. T. Magalhães, and H. D. d. Santos, "A machine learning approach to keystroke dynamics based user authentication," 2007.

[54] E. Bursztein, M. Martin, and J. Mitchell, "Text-based captcha strengths and weaknesses," in *Proceedings of the 18th ACM conference on Computer and communications security*, pp. 125–138, ACM, 2011.

[55] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in *Networking, sensing and control, 2004 IEEE international conference on*, vol. 2, pp. 749–754, IEEE, 2004.

[56] E. Prouff, R. Strullu, R. Benadjila, E. Cagli, and C. Dumas, "Study of deep learning techniques for side-channel analysis and introduction to ascad database.," *IACR Cryptology ePrint Archive*, vol. 2018, p. 53, 2018.

[57] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *arXiv preprint arXiv:1712.07107*, 2017.

[58] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[59] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE, 2017.

[60] "Total malware."

[61] A. Calleja, A. Martín, H. D. Menéndez, J. Tapiador, and D. Clark, "Picking on the family: Disrupting android malware triage by forcing misclassification," *Expert Systems with Applications*, vol. 95, pp. 113–126, 2018.

[62] K. S. Han, J. H. Lim, B. Kang, and E. G. Im, "Malware analysis using visualized images and entropy graphs," *International Journal of Information Security*, vol. 14, no. 1, pp. 1–14, 2015.

[63] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," *arXiv preprint arXiv:1702.05983*, 2017.

[64] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas, "Malware detection by eating a whole exe," *arXiv preprint arXiv:1710.09435*, 2017.

[65] O. Suciu, S. E. Coull, and J. Johns, "Exploring adversarial examples in malware detection," *arXiv preprint arXiv:1810.08280*, 2018.

[66] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," *arXiv preprint arXiv:1809.02077*, 2018.

[67] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[68] I. Homoliak, M. Teknos, M. Ochoa, D. Breitenbacher, S. Hosseini, and P. Hanacek, "Improving network intrusion detection classifiers by non-payload-based exploit-independent obfuscations: An adversarial approach," *arXiv preprint arXiv:1805.02684*, 2018.

[69] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 workshop*, vol. 62, pp. 98–105, Madison, Wisconsin, 1998.

[70] D. Lowd and C. Meek, "Good word attacks on statistical spam filters.," in *CEAS*, vol. 2005, 2005.

[71] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao, "Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pp. 239–254, 2014.

[72] P. Negi, A. Sharma, and C. Robustness, "Adversarial machine learning against keystroke dynamics," 2017.

[73] M. F. Zeager, A. Sridhar, N. Fogal, S. Adams, D. E. Brown, and P. A. Beling, "Adversarial learning in credit card fraud detection," in *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 112–116, IEEE, 2017.

[74] A. Nayebi and S. Ganguli, "Biologically inspired protection of deep networks from adversarial attacks," *arXiv preprint arXiv:1703.09202*, 2017.

[75] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, IEEE, 2016.

[76] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[77] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.

[78] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[79] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 135–147, ACM, 2017.

[80] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[81] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *arXiv preprint arXiv:1710.10766*, 2017.

[82] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu, H. Zhu, and B. Li, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach," *computers & security*, vol. 73, pp. 326–344, 2018.

[83] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," in *International workshop on multiple classifier systems*, pp. 350–359, Springer, 2011.

[84] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi, "Deepcloak: Masking deep neural network models for robustness against adversarial samples," *arXiv preprint arXiv:1702.06763*, 2017.

[85] W. Xu, Y. Qi, and D. Evans, "Automatically evading classifiers," in *Proceedings of the 2016 Network and Distributed Systems Symposium*, pp. 21–24, 2016.

[86] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?," in *International Conference on Machine Learning*, pp. 1689–1698, 2015.

[87] H. S. Anderson, J. Woodbridge, and B. Filar, "Deepdga: Adversarially-tuned domain generation and detection," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, pp. 13–21, ACM, 2016.

[88] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," in *Advances in Neural Information Processing Systems*, pp. 5014–5026, 2018.

[89] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.